8H

УДК 519.876.5

Б.Б. Нестеренко, М.А. Новотарский
Институт математики НАН Украины, г. Киев, Украина
model@imath.kiev.ua

Программная система моделирования сложных физических процессов на клеточных нейронных сетях

В статье приведено обоснование необходимости применения параллельных методов и средств для моделирования сложных физических процессов. Предложено использовать параллельный локально-асинхронный метод решения уравнений математической физики, ориентированный на эффективную реализацию в параллельной вычислительной среде со структурой трехмерного послойного циркулянтного графа. Приведено формальное двухуровневое сетевое описание модели клеточных нейронных сетей со структурой циркулянтного графа для моделирования таких процессов. Описана структура программной системы моделирования, построенная по данному формальному описанию. Кратко описаны функции пакетов и порядок их взаимодействия.

Введение

Моделирование сложных физических процессов, то есть процессов, описываемых уравнениями математической физики, в большинстве случаев является весьма нетривиальной задачей. В особенности если упомянутые процессы протекают в трехмерных областях сложной формы с подвижными границами. Зачастую при построении и исследовании подобных моделей приходится допускать существенные упрощения для получения приемлемого времени прогона модели. При этом неминуемо снижается уровень подобия модели, а следовательно, и практическая ценность результатов моделирования. Современным подходом, позволяющим достигать некоторого прогресса в повышении допустимой сложности математических моделей, является развитие параллельных методов и средств. Вместе с тем параллельным методам решения уравнений математической физики присуща общая черта – ориентация на средства реализации. Поэтому от степени адаптации параллельного метода к той или иной параллельной вычислительной структуре существенно зависит результирующий эффект от его использования. В рамках такого подхода разработан локально-асинхронный метод решения уравнений математической физики на нейронных сетях [1]. Показано, что реализация данного метода наилучшим образом адаптирована к вычислительным средствам со структурой послойного циркулянтного графа [2]. В то же время самым доступным современным параллельным вычислительным средством являются кластерные вычислительные системы с неоднородной структурой межпроцессорных связей. Следовательно, для эффективной реализации разработанных локально-асинхронных методов на кластерных системах необходимо разработать вычислительную среду, представляющую собой программную систему моделирования вычислений на циркулянтном графе.

Целью данной статьи является разработка программной системы моделирования, использующей модель клеточных нейронных сетей со структурой циркулянтного

графа для эффективной реализации вычислительного процесса решения уравнений математической физики локально-асинхронным методом на кластерных вычислительных системах.

Формальное описание модели клеточных нейронных сетей

Для решения трехмерных краевых задач математической физики будем использовать клеточные нейронные сети, структура которых изоморфна трехмерным циркулянтным графам. Рассмотрим применение такого подхода на примере простейшей из таких структур, содержащей 64 нейрона и представленной графом $G(64;1,4,16)$ с шагами по координатам (x, y, z), совпадающими с шагами вдоль колец циркулянтного графа. Таким образом, шаги $(x \pm 1) \mathrm{mod} 16$ соответствуют шагам вдоль кольца наименьшего диаметра, кольцо среднего диаметра образует совокупность вершин, которые могут быть достигнуты путем выполнения шагов $(y \pm 4) \mathrm{mod} 16$, а шаги $(z \pm 16) \mathrm{mod} 64$ совпадают с шагами вдоль кольца с самым большим диаметром. Трехмерная структура, показанная на рис. 1, характеризуется длиной ребра $n = 2^2$ и дает представление о взаимном пространственном размещении нейронов.

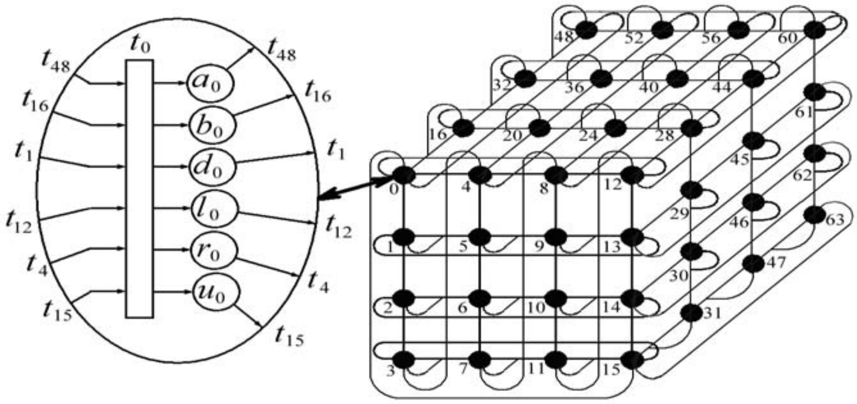


Рисунок 1 – APRO-сетевая модель трехмерной клеточной нейронной сети

Доказано существование подобных структур произвольного размера при условии, что количество нейронов, размещенных вдоль одного ребра, равняется числу, эквивалентному 2^n, где $n = 2,3,4,...$ [2]. Поэтому такой подход дает возможность выбора оптимальной структуры для обеспечения точности модели сложных физических процессов, с одной стороны, и имеющихся вычислительных ресурсов – с другой.

Особенности такой среды состоят в том, что она должна быть представлена с помощью параллельных формальных средств. Только тогда возникает уверенность в сохранении подобия модели при переходе от формального описания к программной реализации [3].

Структуру модели клеточной нейронной сети, показанной на рис. 1, представим APRO-сетью в виде кортежа [4]:

$$\Phi = (P, T, F, M, V),$$

где $P = \left\{ (u_i, r_i, d_i, a_i, b_i) \middle| i = \overline{0,63} \right\}$ – множество позиций;

$T = \left\{ t_i \,\middle|\, i = \overline{0,63} \right\}$ – множество переходов;

$$F = \left\{ \begin{matrix} (u_i, t_{(i-16)\bmod 64}), & (d_i, t_{(i+16)\bmod 64}), \\ (r_i, t_{(i+1)\bmod 16}), & (l_i, t_{(i-1)\bmod 16}), \\ (a_i, t_{(i+4)\bmod 16}), & (b_i, t_{(i-4)\bmod 16}) \end{matrix} \right\}_{i=0}^{63} \cup \left\{ \begin{matrix} (t_i, d_{(i-16)\bmod 64}), & (t_i, u_{(i+16)\bmod 64}), \\ (t_i, l_{(i+1)\bmod 16}), & (t_i, r_{(i-1)\bmod 16}), \\ (t_i, b_{(i+4)\bmod 16}), & (t_i, a_{(i-4)\bmod 16}) \end{matrix} \right\}_{i=0}^{63} -$$

множество ребер между переходами и позициями;

$$M = \left\{ \begin{matrix} \left(u_i, \left\{ \mu_j^U \right\}_{j=1}^{Max_u_i} \right), & \left(d_i, \left\{ \mu_j^D \right\}_{j=1}^{Max_d_i} \right), \\ \left(r_i, \left\{ \mu_j^R \right\}_{j=1}^{Max_r_i} \right), & \left(l_i, \left\{ \mu_j^l \right\}_{j=1}^{Max_l_i} \right), \\ \left(a_i, \left\{ \mu_j^A \right\}_{j=1}^{Max_a_i} \right), & \left(b_i, \left\{ \mu_j^B \right\}_{j=1}^{Max_b_i} \right) \end{matrix} \right\}_{i=0}^{63}$$ – конечное множество маркирований;

$V = (\Delta, \Psi, \Lambda)$ – множество глобальных переменных: Δ − множество показателей производительности, Ψ – множество показателей реактивности, Λ – показатели использования программных и аппаратных ресурсов.

Структура APRO-сети, включающая упомянутые элементы, образует иерархический уровень 0 формального описания модели клеточной нейронной сети. Уровень 1 модели раскрывает APRO-сетевую структуру клеточного нейрона, представленного операторным переходом t_i [4]. Таким образом, переходы APRO-сети на уровне 0 представляют клеточные нейроны, а на уровне 1 включают набор процедур, моделирующих работу элементов клеточного нейрона. Позиции на уровне 0 отображают структуры данных, обслуживающие каналы межнейронного обмена информацией. На уровне 1 переходы представляют структуры данных, участвующих во внутренней обработке. Множество ребер на уровне 0 отображает структуру межнейронных связей, а на уровне 1 – структуру связей между элементами клеточного нейрона. На рис. 2 показана APRO-сетевая модель, описывающая нейрон на уровне 1 в виде операторного перехода t_i.

Кортеж APRO-сети, описывающий операторный переход нейрона:

$$t_i = (P_i, T_i, E_i, X_i, F_i),$$

где $P_i = \left\{ p_{i,j} \right\}_{j=1}^{18}$ – множество позиций; $T_i = \left\{ \tau_{i,q} \right\}_{q=1}^{24}$ – множество переходов;

$E_i = \left\{ e_{i,c} \right\}_{c=1}^{6}$ – множество входов; $X_i = \left\{ x_{i,d} \right\}_{d=1}^{6}$ – множество выходов;

$F_i = (P_i \times T_i) \cup (T_i \times P_i) \cup (E_i \times T_i) \cup (T_i \times X_i)$ – множество ребер.

Функционирование данной модели клеточной нейронной сети формально представлено в виде последовательности процессов ввода, обработки информации и процесса локального взаимодействия [5]:

$$P := A.B.O.$$

Процесс ввода A является сложным процессом, отображающим множество процессов ввода рабочей нагрузки в каждый нейрон. В случае однотипного задания рабочей нагрузки процесс A может быть представлен множеством клонированных процессов $A = \overbrace{\{A_0, A_0, ..., A_0\}}^{N}$, каждый из которых описывает ввод информации в отдельный узел клеточной нейронной сети. Если же каждый клеточный нейрон получает

индивидуальную рабочую нагрузку, то процесс A рассматривается как параллельная композиция процессов $A = A_0 | A_1 | ... | A_i | ... | A_{N-1}$, где A_i – процесс ввода информации в клеточный нейрон t_i. В общем случае ввод рабочей нагрузки в клеточную нейронную сеть допускает существование определенных подмножеств нейронов, характеризующихся идентичными рабочими нагрузками. В этом случае процесс A будет иметь вид:

$$A = \overbrace{\{A_0, ..., A_0\}}^{k_0} |...| \overbrace{\{A_i, ..., A_i\}}^{k_i} |...| \overbrace{\{A_{n-1}, ..., A_{n-1}\}}^{k_{n-1}},$$

где k_i – количество нейронов, входящих в i-е подмножество; A_i – процесс ввода рабочей нагрузки, идентичный для всех k_i нейронов i-го подмножества; n – количество подмножеств при общем количестве нейронов клеточной нейронной сети: $N = \sum_{i=0}^{n-1} k_i$.

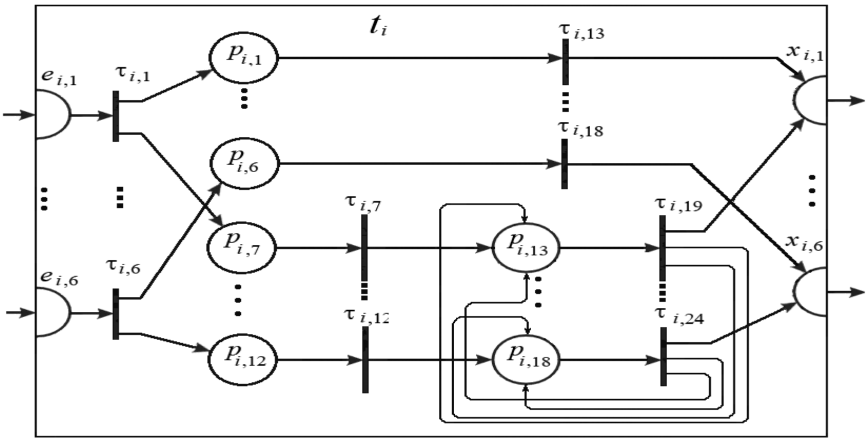


Рисунок 2 – APRO-сетевая модель нейрона трехмерной клеточной нейронной сети

Процесс обработки информации B состоит из множества процессов

$$B = \{B_i | i = 0, \cdots N-1\},$$

элемент которого B_i отображает процесс обработки информации в отдельном нейроне. В общем случае этот процесс представим выражениями алгебры процессов [5], задающими правила асинхронного взаимодействия каждого нейрона клеточной сети с параллельной композицией, в которую входят остальные нейроны данной сети. Произвольный процесс обработки информации B_i в нейроне t_i состоит из двух процессов, отображающих основные функции нейрона – вычислительную и коммуникационную. Во время функционирования нейрона упомянутые процессы взаимодействуют между собой, образуя общий процесс B_i. Входной коммуникационный процесс включает множество процессов, обеспечивающих анализ входных данных с целью сепарации транзитной информации:

$$K_i^{in} = \{K_i^j | j = 1, ..., m\},$$

где m – количество каналов связи клеточного нейрона t_i.

Если результат анализа соответствующего канала указывает на то, что полученные данные являются транзитной информацией, то запускается процесс $W_i^j \subset K_i^j$

для определения дальнейшего маршрута путем выбора соответствующего выходного канала связи. Если данные, полученные по входному каналу, предназначенные для нейрона t_i, то происходит передача данных вычислительному процессу $Q_i^j \subset K_i^j$. Результатом обоих действий является передача данных через выходные каналы связи.

Процесс локального взаимодействия включает множество процессов

$$O = \{O_i \mid i := 0,...,N-1\},$$

каждый из которых представлен процессами асинхронного взаимодействия клеточного нейрона t_i с соседними нейронами $O_i = \{O_i^j \mid j = 1,...,m\}$. Правила такого взаимодействия отображают цикличный асинхронный характер межнейронного обмена информацией.

Структура программной системы моделирования

Исследования показали, что для решения реальных практических задач модель клеточной нейронной сети должна состоять минимально из десятков тысяч нейронов. Очевидно, что такая модель ориентирована на вычислительные ресурсы, существенно превышающие возможности персональных компьютеров. Наиболее доступными и динамически развивающимися параллельными вычислительными средствами на сегодня являются кластерные системы, что и определило их выбор в качестве средства реализации данного типа моделей. В связи с этим возникает проблема равномерной загрузки узлов кластерной системы и распределения коммуникационных потоков данных при реализации модели клеточной нейронной сети со структурой трехмерного послойного циркулянтного графа. Эта проблема решена с помощью разработки программной системы моделирования, структура которой представлена множеством пакетов со связями, показанными на рис. 3.

Ядром программной системы моделирования является пакет классов ModelingCore, создаваемый первым, и выполняющий функцию «входной точки» программной системы моделирования. ModelingCore создает пакет классов MainGUI (окна и диалоги) для общения с пользователем.

Главный интерфейс MainGUI содержит команды системной настройки. В частности, с помощью этого интерфейса пользователь инициирует создание пакета классов APRONetBuilder, главная цель которого – сгенерировать структурное представление APRO-сети. APRONetBuilder создает APROBuilderGUI, через который пользователь может задавать модель. Структурная модель APRO-сети формируется непосредственно путем использования графических элементов интерфейса APROBuilderGUI в случае, когда количество ее объектов не превышает 128. При построении моделей с большим количеством объектов или моделей со сложной структурой связей между переходами и позициями используют алгебру процессов. Для этого финальная команда задания модели «build» запускает процесс парсинга. Когда пользователь инициирует генерацию сети, APRONetBuilder создает промежуточный строитель ProcessAlgebraBuilder. С помощью ProcessAlgebraBuilder создается TextParser, который включает интерфейс ProcessAlgebraGUI. При этом возникает такой поток выполнения: с помощью ProcessAlgebraGUI формируется скрипт описания модели, который получает TextParser и формирует из него дерево разбора. Упомянутое дерево разбора передается в ProcessAlgebraBuilder. Пакет ProcessAlgebraBuilder раскрывает циклы, разветвления, то есть «выполняет» дерево разбора и формирует на выходе дерево выражений алгебры процессов, которое отдает в APRONetBuilder.



Рисунок 3 – Структура программной системы моделирования

Таким образом реализуется одно из важных отличий данной программной системы моделирования, заключающееся в комбинированном подходе, объединяющем процессное моделирование и моделирование списком событий. APRONetBuilder уже из дерева выражений алгебры процессов по заданным правилам формирует APRONet, включающий в себя APRONetCalcNodes и APRONetComNodes (переходы и вершины) – реализации общего интерфейса APRONetNodes. От каждого APRONetNodes есть связь с пакетом классов WorkLoad, откуда формируются «реальные» процедуры перехода. Сначала эти связи пусты, и построенный APRONet возвращается в ModelingCore, который с помощью GUI информирует пользователя об успехе построения структуры сети (или информирует об ошибках в синтаксисе, в выражениях алгебры процессов и т.п.).

После этого ModelingCore создает APRONetConfigurator и передает ему ранее построенный APRONet. Конфигуратор создает APRPConfigurationGUI – для приема от пользователя параметров конфигурации (области моделирования, параметров метода и т.п.). На основании этой информации конфигуратор устанавливает внутренние параметры вершин в APRONet. Сразу же создаются конечные экземпляры классов реальной рабочей нагрузки и присоединяются к вершинам сети.

После этого готовая сеть передается в APROModelExecutor – набор классов, обеспечивающих выполнение модели. Интерфейс ModelExecutorGUI – интерфейс квалифицированного пользователя для тонкой настройки параметров системы выполнения модели. Интерфейс ModelResultGUI предназначен для отображения результатов моделирования.

Интерфейс RealCalculationNode – отдельный пакет, подлежащий выполнению на физическом узле кластера. Фактически экземпляры программ, созданных на основе данного пакета, передаются в очередь на кластер и размещаются по физическим узлам в соответствии с инструкциями локального менеджера ресурсов. Каждый RealCalculationNode связан с помощью TCP-сокета с Dispatcher. Пакет Dispatcher лежит в основе программы, запущенной на главном узле (MasterNode). Dispatcher играет роль посредника между вычислительными узлами и APROModelExecutor. Пакет APROModelExecutor системы моделирования передает в Dispatcher готовую к выполнению сеть. Dispatcher распределяет виртуальные узлы сети по реальным узлам RealCalculationNode. Пакеты RealCalculationNode для этого создают внутренние классы VirtualCalculationNode, инкапсулирующие переходы APRO-сети. InternalDataNode инкапсулируют вершины, общие для переходов внутри реального узла, ExternalDataNode инкапсулируют вершины, связанные с переходами в других узлах, и потому требуют сетевой передачи данных. VirtualCalculationNode выполняют рабочую нагрузку, RealCalculationNode перераспределяет данные между InternalDataNode, а также отвечает за передачу данных между своими ExternalDataNode и ExternalDataNode других узлов (или через диспетчер, или через прямые каналы, установленные с помощью диспетчера). Результаты моделирования передаются диспетчеру, а через него – в APROModelExecutor для графического отображения. После завершения вычислений RealCalculationNode завершают работу и освобождают очередь кластера.

Выводы

В работе приведено обоснование необходимости использования параллельных методов и средств для моделирования сложных физических процессов, описываемых уравнениями математической физики. Поскольку параллельный локально-асинхронный метод решения уравнений математической физики предполагает использование параллельной вычислительной среды со структурой трехмерного послойного циркулянтного графа, для эффективной его реализации на кластерных системах предложена программная система моделирования. В основу данной системы моделирования положена модель клеточной нейронной сети со структурой циркулянтного графа. Приведено ее формальное описание с использованием APRO-сетей. Для описания принципов функционирования данной модели использована алгебра процессов.

Общая структура программной системы моделирования представлена в виде UML-диаграммы пакетов. Кратко описаны функции пакетов и порядок их взаимодействия.

Литература

1. Новотарский М.А. Искусственные нейронные сети: вычисления / М.А. Новотарский, Б.Б. Нестеренко. – К. : Институт математики НАН Украины, 2004. – 408 с.
2. Нестеренко Б.Б. Клеточные нейронные сети на циркулянтных графах / Б.Б. Нестеренко, М.А. Новотарский // Искусственный интеллект. – 2009. – № 3. – С. 132-138.

3. Ермаков С.М. Математическая теория планирования эксперимента / Ермаков С.М. – М. : Наука, 1983. – 392 с.
4. Нестеренко Б.Б. Дискретные клеточные нейронные сети / Б.Б. Нестеренко, М.А. Новотарский // Научная сессия МИФИ – 2008. X Всероссийская научно-техническая конференция “Нейроинформатика – 2008”», (22 – 25 января 2008 г.) : сборник научных трудов. – М. : МИФИ, 2008. – Ч. 2. – С. 31-39.
5. Нестеренко Б.Б. Алгебра процессов для моделирования параллельных асинхронных вычислительных структур / Б.Б. Нестеренко, М.А. Новотарский // Электронное моделирование. – 2006. – Т. 28, № 4. – С. 47-64.

Literatura

1. Novotarskij M.A. Iskusstvennye nejronnye seti: vychislenija. K.: Institut matematiki NAN Ukrainy. 2004. 408 s.
2. Nesterenko B.B. Iskusstvennyj intellekt. 2009. № 3. S. 132-138.
3. Ermakov S.M. Matematicheskaja teorija planirovanija jeksperimenta. M. : Nauka. 1983. 392 s.
4. Nesterenko B.B. Nauchnaja sessija MIFI. 2008. X Vserossijskaja nauchno-tehnicheskaja konferencija “Nejroinformatika – 2008”, (22 – 25 janvarja 2008 g.) Sbornik nauchnyh trudov. M. : MIFI. 2008. Ch. 2. S. 31-39.
5. Nesterenko B.B. Jelektronnoe modelirovanie. 2006. T. 28. № 4. S. 47-64.

Б.Б. Нестеренко, М.А. Новотарський

Програмна система моделювання складних фізичних процесів на клітинних нейронних мережах

У статті наведено обґрунтування необхідності застосування паралельних методів та засобів для моделювання складних фізичних процесів. Запропоновано використовувати паралельний локально-асинхронний метод розв’язування рівнянь математичної фізики, орієнтований на ефективну реалізацію в паралельному обчислювальному середовищі зі структурою тривимірного пошарового циркулянтного графа. Наведено формальний дворівневий мережний опис моделі клітинних нейронних мереж зі структурою циркулянтного графа для моделювання таких процесів. Описана структура програмної системи моделювання, яка побудована за даним формальним описом. Коротко описані функції пакетів і порядок їх взаємодії.

B.B. Nesterenko, M.A. Novotarskiy

Software System for Simulation of Complex Physical Processes on Cellular Neural Networks

In this paper rationale for the use of parallel methods and tools for simulation of complex physical processes is given. It is offered to use a parallel locally-asynchronous method for solving equations of mathematical physics, oriented on effective implementation in parallel computing environment with structure of three-dimensional layer wise circulant graph. A formal description of a two-tier model of cellular neural networks with the structure of circulant graph to simulate these processes is given. The structure of software system for modeling, created according to the formal description, is described. Functions of packages and an order of their interaction are shortly described.

Статья поступила в редакцию 04.07.2011.